Eclipse Plug-in Development Research Project

Report 12

UNIVERSITY AT BUFFALO

September 2, 2015

Authored by: Chern Yee Chua

Eclipse Plug-in Development Research Project

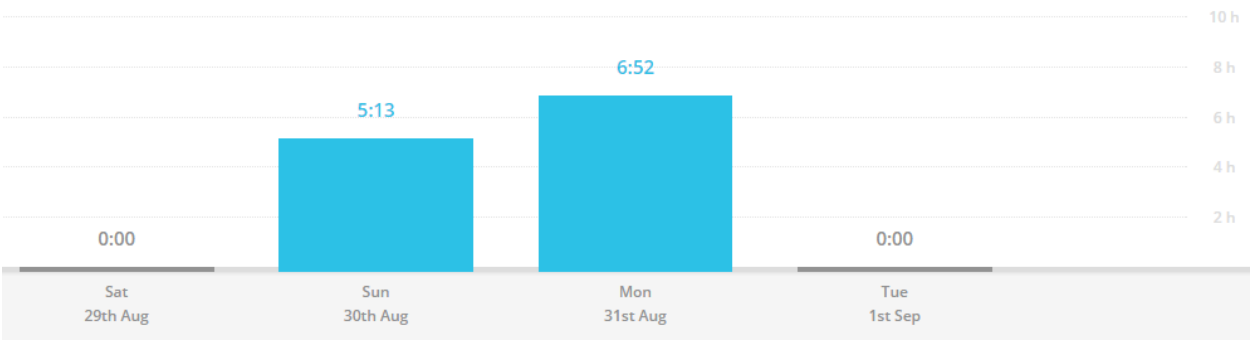
Report 12

Introduction

In this week, I have fixed the issue where changing files or projects will print out the entire AST node to only initialize once when eclipse starts.

Synopsis

I spend around 10 hours (actually much more than that) on this project for this week.



Eclipse Plugin	12:05:00
adding the child block range so that the output is clearer	6:52:00
fixing the printing everytime CU is changed	5:13:00

Explanation of content

The initial content will no longer be printing out every time the CompilationUnit is changed. Instead, it will only be initialized once when the eclipse starts. In that sense, the size of recorded file will be reduced significantly.

The other thing that I have added in is the block section. By looking at the output of recorded file, we might not know the scope of variables, methods, and things like that. So, I decided to add the scope of certain block, such as if else statement block, try catch block, and method block. See the example below:

```
[LINE INITIALIZED (gddfg)]: 17 $ [VARIABLE_DECLARATION_STATEMENT] (int) | (z) |
[LINE INITIALIZED (gddfg)]: 19 $ [IF_STATEMENT] booleanValue (true) |
[LINE INITIALIZED (gddfg)]: 20 $ [ASSIGNMENT] (x) | operator (=) | token (0) |
[LINE INITIALIZED (gddfg)]: 21 $ [ASSIGNMENT] (y) | operator (=) | token (0) |
[LINE INITIALIZED (gddfg)]: 22 $ [ASSIGNMENT] (z) | operator (=) | token (0) |
[LINE INITIALIZED (gddfg)]: 28 $ [VARIABLE_DECLARATION_STATEMENT] (int) | (ww) |
[LINE INITIALIZED (gddfg)]: 29 $ [VARIABLE_DECLARATION_STATEMENT] (int) | (yy) |
[LINE INITIALIZED (gddfg)]: 30 $ [VARIABLE_DECLARATION_STATEMENT] (int) | (zz) |
[LINE INITIALIZED (gddfg)]: 34 $ [VARIABLE_DECLARATION_STATEMENT] (int) | (aa) |
[LINE INITIALIZED (gddfg)]: 35 $ [VARIABLE_DECLARATION_STATEMENT] (int) | (bb) |
[LINE INITIALIZED (gddfg)]: 38 $ [SIMPLE_TYPE] (Exception) | varargs (false) | (e) |
[LINE INITIALIZED (gddfg)]: 39 $ [VARIABLE_DECLARATION_STATEMENT] (int) | (zz) |
```

By looking at the output, you might be confused of what the actual code looks like. Here's what I have added to make the output more detailed:

```
[LINE INITIALIZED (gddfg)]: 17 $ [VARIABLE_DECLARATION_STATEMENT] (int) | (z) |
[LINE INITIALIZED (gddfg)]: 19 $ [IF_STATEMENT] booleanValue (true) |
[LINE INITIALIZED (gddfg)]: 19 $ OPEN [IF_STATEMENT]
[LINE INITIALIZED (gddfg)]: 20 $ [ASSIGNMENT] (x) | operator (=) | token (0) |
[LINE INITIALIZED (gddfg)]: 21 $ [ASSIGNMENT] (y) | operator (=) | token (0) |
[LINE INITIALIZED (gddfg)]: 22 $ [ASSIGNMENT] (z) | operator (=) | token (0) |
[LINE INITIALIZED (gddfg)]: 22 $ CLOSE [IF_STATEMENT]
[LINE INITIALIZED (gddfg)]: 19 $ OPEN [ELSE_STATEMENT]
[LINE INITIALIZED (gddfg)]: 28 $ [VARIABLE_DECLARATION_STATEMENT] (int) | (ww) |
[LINE INITIALIZED (gddfg)]: 29 $ [VARIABLE_DECLARATION_STATEMENT] (int) | (yy) |
[LINE INITIALIZED (gddfg)]: 30 $ [VARIABLE_DECLARATION_STATEMENT] (int) | (zz) |
[LINE INITIALIZED (gddfg)]: 30 $ CLOSE [ELSE_STATEMENT]
[LINE INITIALIZED (gddfg)]: 33 $ OPEN [TRY_STATEMENT]
[LINE INITIALIZED (gddfg)]: 34 $ [VARIABLE_DECLARATION_STATEMENT] (int) | (aa) |
[LINE INITIALIZED (gddfg)]: 35 $ [VARIABLE_DECLARATION_STATEMENT] (int) | (bb) |
[LINE INITIALIZED (gddfg)]: 35 $ CLOSE [TRY_STATEMENT]
[LINE INITIALIZED (gddfg)]: 38 $ [SIMPLE_TYPE] (Exception) | varargs (false) | (e) |
[LINE INITIALIZED (gddfg)]: 38 $ OPEN [CATCH_CLAUSE]
[LINE INITIALIZED (gddfg)]: 39 $ [VARIABLE_DECLARATION_STATEMENT] (int) | (zz) |
[LINE INITIALIZED (gddfg)]: 39 $ CLOSE [CATCH_CLAUSE]
[LINE INITIALIZED (gddfg)]: 39 $ CLOSE [METHOD_DECLARATION]
```

And now you can see that there is not only if statement, but also else statement and a try statement, what a surprise!

Conclusion

The latest implementation (Version 1.0.4a) might be a bit unstable (so far I have fixed most of the bugs from what I have tested but more testing is needed) but the previous (Version 1.0.4) is working properly.

All I can say is the plugin is doing a better job than before. The data that we collect is more intelligent and more accurate, and various issues have been handled.

Thank you for your time!